ISSN 2007-9737

A Novel Cooperative Hybrid Metaheuristic Optimization Method Based on Collective Intelligence

Hector Martinez, Fevrier Valdez*

Tijuana Institute Technology, TecNM,
Mexico

fevrier@tectijuana.mx

Abstract. This paper studies a cooperative hybrid metaheuristic that combines the Dragonfly Algorithm (DA), the Firefly Algorithm (FA) and Cuckoo Search (CS) in a multi-population setting with fuzzy-based parameter adaptation. Each algorithm keeps its own update rules, while a shared global best solution allows information exchange among subpopulations. Type-1, Interval Type-2 and General Type-2 fuzzy controllers adjust key parameters during the run so that exploration and exploitation are modified according to the current progress of the search. The hybrid is tested on a set of standard continuous benchmark functions. The cooperative model without fuzzy logic already improves over the individual algorithms in most cases, and the fuzzy variants further reduce the mean error and make convergence more stable. In particular, the General Type-2 configuration is often the best performer. These results indicate that combining complementary swarm behaviors with fuzzy uncertainty handling is a practical way to implement dynamic parameter control in hybrid metaheuristics.

Keywords. Hybrid metaheuristic, cooperative multi-population optimization, convergence analysis.

1 Introduction

During the last decade, nature-inspired metaheuristics have been widely adopted to solve optimization problems that are difficult for classical deterministic methods, especially when the search space is nonlinear, multimodal or high-dimensional [1]. Instead of relying on strong analytical assumptions, these algorithms imitate simple interaction rules observed in biological systems and use them to explore and exploit the search space. This flexibility has made them a practical option in many engineering and scientific applications where exact models are either unavailable or too costly to use [2].

Within this family, the Dragonfly Algorithm (DA), the Firefly Algorithm (FA) and Cuckoo Search (CS) are well known representatives with complementary search behaviors. DA models swarm interactions through separation, alignment and cohesion, which naturally balance exploration and exploitation [3]. FA is driven by attractiveness based on perceived brightness and is particularly effective at refining candidate solutions around promising regions [4]. CS uses Lévy flights and a parasitic reproduction scheme to generate long jumps in the search space, helping the algorithm escape local optima and maintain diversity [5]. Several studies have shown that these methods can be useful in domains such as biomedical signal processing, feature selection and hyper parameter tuning for classification models [6,7].

However, relying on a single metaheuristic with fixed parameters often leads to inconsistent performance across different problem classes. Previous work on hybridization and parameter adaptation has shown that combining multiple search strategies and adjusting their behavior over time can improve convergence stability and solution quality [8]. At the same time, the No-Free-Lunch theorem reminds us that no algorithm is universally superior, which motivates the design of hybrid frameworks tailored to specific scenarios [9]. Recent contributions have explored DA-based hybrids and fuzzy controllers for parameter tuning, reporting gains in robustness and accuracy on benchmark and application-oriented problems [10,11]. Building on these ideas, this paper proposes a cooperative multi-population hybrid that merges DA, FA and CS under a common framework with fuzzy-driven parameter adaptation. Each algorithm governs its own subpopulation and preserves its original update rules, while a periodic elite-exchange mechanism allows the best individuals to migrate across subgroups. In the context of many objective problems, diversity preservation has been shown to be a key factor for obtaining high quality solution sets, who propose specific measures to assess and maintain diversity during the optimization process [30]. Type-1, Interval Type-2 and General Type-2 fuzzy controllers are used to modify key parameters during the run, with the goal of adjusting the balance between exploration and exploitation according to simple indicators of search progress. The proposed approach is evaluated on standard continuous benchmark functions to examine whether this combination of complementary swarm behaviors and fuzzy uncertainty handling leads to more reliable convergence and better solution quality than standalone implementations. More recently, a wide range of evolutionary algorithms enhanced with extended fuzzy logic systems, showing how these approaches can improve robustness and adaptability when dealing with complex optimization problems [31]. Section 2 outlines the materials and methods adopted in this work. Section 3 presents the experimental results and the analysis. Finally, Section 4 offers the conclusions.

2 Materials and Methods

The approach taken in this study starts by outlining the scenario in which the optimization problems are solved. Once this setup is in place, each algorithm is configured according to its standard formulation. The experiments are then carried out following a procedure that allows us to observe and assess how the methods perform under the same conditions.

2.1 Hybrid Swarm Optimization Framework

The core of the proposed approach is a cooperative hybrid framework that combines three bio-inspired algorithms: the DA, the FA and CS. Hybridization and multi-swarm cooperation have proven effective strategies to improve convergence speed, stability, and robustness across complex landscapes, particularly when different heuristics contribute complementary search behaviors [12-14]. Each algorithm maintains its own population and search dynamics, but they share information through a common global best solution. Every 10 iterations, the hybrid system selects the best candidate among the three sub–populations and uses it as a reference to guide the subsequent updates of DA, FA and CS [15,16].

The three metaheuristics are used in their standard continuous versions: DA models static and dynamic swarming behavior of dragonflies and balances exploration and exploitation through separation, alignment, cohesion, attraction to food, and distraction from enemies [17,18].

FA simulates the attraction among fireflies based on their brightness and is especially suitable for multimodal functions. CS is based on Lévy flights and brood parasitism, providing strong global exploration capabilities [19,20].

In the baseline configuration (without fuzzy logic), each algorithm is run with fixed control parameters taken from the original references. This configuration, is used as a non-adaptive benchmark to evaluate the impact of the proposed fuzzy parameter adaptation schemes.

2.2 Fuzzy-Based Parameter Adaptation

To enhance the balance between exploration and exploitation, the hybrid framework is extended with three fuzzy adaptation schemes: Type-1 Fuzzy Logic (T1-FL), Interval Type-2 Fuzzy Logic (IT2-FL), and General Type-2 Fuzzy Logic (GT2-FL). This leads to four configurations: No fuzzy adaptation (baseline): fixed parameters [22].

Type-1 fuzzy adaptation: classical fuzzy controllers

Interval. Type-2 fuzzy adaptation: IT2 Gaussian controllers [21]. General Type-2 fuzzy adaptation: GT2 z-slice based controllers [23, 27, 28]. In all fuzzy configurations, the same high-level architecture is preserved: three independent fuzzy systems drive the key parameters of DA, FA and CS, respectively [24, 29].

ISSN 2007-9737

2.2.1 VInput and Output Variables

The fuzzy adaptation layer uses three input signals that summarize the state of the search:

$t_{norm} \in [0,1]$: Normalized iteration index, representing the progress of the optimization process:

$$\mathrm{t_{norm}} = \frac{t}{T_{max}}. \tag{1}$$

Which represents the progress of the optimization process. Similar progress-driven adaptation schemes are used in dynamic evolutionary optimization [26].

$D \in [0,1]$: Population diversity, computed from the standard deviation of the agents' positions and normalized by the search range:

$$D = \frac{1}{n}\sum_{j=1}^{n} \frac{\sigma_j}{X_{max} - X_{min}}, \tag{2}$$

Where:

σ_j = desviación estándar de la población en la dimensión j.

n = número de dimensiones.

$I \in [0,1]$: Recent improvement index, measuring the relative fitness improvement over a sliding window of past iterations [25,26]. These inputs are used with different combinations:

DA controller $[\mathrm{t_{norm}}, D_{DA}] \rightarrow$ output: inertia weight w.

FA controller: $[\mathrm{t_{norm}}, D_{FA}, I] \rightarrow$ outputs: αFA (randomization factor) and β_0(attractiveness).

CS controller: $[\mathrm{t_{norm}}, D_{CS}, I] \rightarrow$ outputs: αCS (Lévy step size) and P_a (abandonment probability).

The diversity and improvement metrics follow the general ideas of dynamic parameter adaptation found in recent fuzzy-based optimization works, including those developed by Castillo et al. at, where Type-1, interval Type-2 and generalized Type-2 fuzzy systems have been successfully used to tune metaheuristic parameters in real time.

To ensure numerical stability, the fuzzy outputs are clipped to predefined ranges, showed on Table 1.

These bounds were chosen according to typical values in the literature and preliminary experiments.

Table 1. Predetermined Ranges

Symbol	Values
w	$[0.40, 0.95]$
αFA	$[0.01, 0.35]$
β_0	$[0.60, 1.60]$
αCS	$[0.001, 0.08]$
P_a	$[0.10, 0.35]$

2.2.2 Type-1 Fuzzy Logic (T1-FL)

In the T1 configuration, each controller is implemented as a Mamdani or Sugeno Type-1 fuzzy system with Gaussian membership functions for the input variables. The rules encode simple expert knowledge such as: "If diversity is low and time is high, then reduce exploration and favor exploitation.", "If recent improvement is small, then increase exploration."

The three T1 fuzzy systems are constructed in MATLAB. During execution, they are evaluated at each iteration using the current values of t_{norm}, D and I. This serves as a baseline for comparison with the Type-2 schemes and follows the general approach of Type-1 based dynamic parameter tuning reported in fuzzy optimization literature.

2.2.3 Interval Type-2 Fuzzy Logic (IT2-FL)

In the IT2 configuration, the Type-1 membership functions are extended to interval Type-2 Gaussian MFs. For each linguistic term, an upper and a lower Gaussian function are defined, with the same center but slightly different standard deviations, creating a Footprint of Uncertainty (FOU). Each IT2 fuzzy controller is implemented as a pair of T1 systems (upper and lower). The inference and Type-reduction are carried out using a simplified interval centroid. This follows the line of research on interval Type-2 fuzzy systems for parameter adaptation, where IT2 controllers have shown superior robustness under uncertainty.

2.2.4 General Type-2 fuzzy logic (GT2-FL)

The GT2 configuration aims to capture more complex uncertainty patterns by approximating General Type-2 fuzzy systems through a z-slice based scheme. Instead of explicitly modeling a continuous secondary membership function, the approach generates several perturbed versions of the original T1 FIS (z-slices), each obtained by scaling the standard deviations of the Gaussian membership functions by a factor related to the z level. For each input vector, all slices are evaluated and their outputs are aggregated through a weighted average, which acts as an approximate Type-reduction step.

This methodology is inspired by recent work on generalized Type-2 fuzzy logic in swarm and evolutionary optimization, where GT2 systems have been shown to improve robustness and performance in noisy or highly nonlinear scenarios. In this way, the proposed GT2 fuzzy adaptation extends the IT2 scheme by introducing an additional degree of flexibility in the modeling of uncertainty, while remaining computationally feasible for iterative optimization.

2.3 Benchmark Functions

The performance of the four configurations is evaluated on ten continuous benchmark functions frequently used in the literature for testing metaheuristic optimizers. These include unimodal and multimodal landscapes, with varying degrees of separability and conditioning (e.g., Sphere, Rosenbrock, Rastrigin, Ackley, Schwefel-Type functions), covering both simple and highly rugged search spaces. For all experiments, the following protocol is used:

Dimensionality: each benchmark function is evaluated in 1000 dimensional search space.

Search domain: lower and upper bounds are set according to the standard definitions of each function.

Population size: 40 agents per algorithm (DA, FA, CS).

Stopping criterion: a maximum of 1000.

Number of runs: each configuration is executed independently for a fixed number of 30 runs using different random seeds to obtain statistically meaningful results. Performance metrics: for each benchmark, the best, mean, median, and standard deviation of the best-found fitness values are reported over all runs; convergence curves (best cost versus iteration) are also analyzed.

The algorithmic structure and parameter settings are consistent across the four configurations, so that the only difference between them is the presence and Type of fuzzy adaptation layer. This experimental design allows an objective comparison of the impact of Type-1, Interval Type-2, and General Type-2 fuzzy logic on the convergence behavior and solution quality of the hybrid DA–FA–CS framework.

Table 2. Parameters adapted by fuzzy logic

Algorithm	Parameter	Symbol
Dragonfly Algorithm (DA)	Inertia weight	w
Firefly Algorithm (FA)	Randomization factor	αFA
Firefly Algorithm (FA)	Base attractiveness	β_0
Cuckoo Search (SC)	Levy step size	αCS
Cuckoo Search (CS)	Abandonment probability	P_a

2.4 Architecture and Integration

To avoid modifying the internal structure of DA, FA and CS, the fuzzy layer is integrated by means of a global data exchange mechanism. At each iteration, the fuzzy controllers compute the updated parameters: w, αFA, β_0, αCS and p_a. Showed in Table 2.

This design keeps the signatures of the step functions unchanged and allows switching between the four configurations (no fuzzy, T1, IT2, GT2). It also facilitates future extensions, such as adding new fuzzy schemes or hybridizing with other swarm optimizers, without rewriting the core algorithms.

The proposed optimization framework combines three nature-inspired algorithms DA, FA and CS into a unified, cooperative multi-swarm architecture. Instead of running each algorithm independently or sequentially, the framework allows the three swarms to coexist and evolve in parallel, exchanging information through a shared global-best solution. This design aims to exploit the distinct strengths of each swarm while compensating for their individual weaknesses, producing a more stable and adaptive search behavior.

Each swarm preserves its own population structure, position-update rules, and internal mechanism for balancing exploration and exploitation. However, after each iteration, all swarms report their best candidate solution, and the best among them is designated as the global best. This global best is then broadcast back to all three algorithms, influencing their subsequent movements and encouraging cooperation. In this way, the hybrid framework acts as a supervisory mechanism that coordinates multiple search heuristics without altering their fundamental equations.

Complementarity of Search Dynamics

The motivation for integrating DA, FA, and CS is rooted in the complementary nature of their search operators:

In the hybrid environment, DA contributes: Structured movement patterns, group-coordinated exploitation and smooth convergence behavior. Because DA is sensitive to parameter settings, its performance improves substantially when guided by the fuzzy adaptation module.

FA contributes: Strong local search capability, self-adaptive attractiveness fields and rapid refinement near promising regions. FA often converges quickly but may become trapped in local minima. In the hybrid system, CS's global exploration and DA's social dynamics mitigate this weakness.

CS contributes: High exploratory power, ability to escape deep local optima and long jumps enabling global coverage. This complements FA’s local search and DA’s structured movement, preventing premature convergence.

At each iteration, the following cooperation occurs:

Independent update: DA, FA, and CS each evaluate and update their populations using their native equations.

Local best selection: Each algorithm identifies its own best candidate.

Cross-swarm comparison: The best among all three best candidates is selected as the hybrid global best.

Broadcast mechanism: This global best is shared with all algorithms and inserted into the next iteration’s update logic (e.g., DA’s food source, FA’s brightest firefly, CS’s best nest).

Feedback loop: The updated fuzzy controllers use swarm behavior metrics to adapt algorithm-specific parameters dynamically.

This cooperative mechanism ensures that: FA fast local refinement benefits from CS global jumps. DA structured movement stabilizes the overall behavior. The three algorithms do not stagnate in the same region simultaneously.

The final result is a search engine that is more resilient, adaptive, and capable of maintaining diversity compared with any individual algorithm run in isolation.

2.4.1 Dragonfly Algorithm Setup

The DA exploration and exploitation balance is influenced by the inertia weight, the inertia weight w in the DA plays a crucial role in regulating the balance between exploration (global search) and exploitation (local refinement). By controlling the influence of the previous movement vector, the inertia weight modifies how aggressively or conservatively dragonflies move across the search landscape. Its benefits are summarized as follows and that’s the parameter that we optimized on this hybrid.

In the classical formulation, it decreases linearly:

$$w(t) = w_{max} - t_{norm}(w_{max} - w_{min}), \tag{3}$$

Under fuzzy control, the parameter becomes:

$$w(t) = FIS_{DA}(t_{norm}, D_{DA}), \tag{4}$$

Subject to the operational range:

$$w(t) \in [0.40, 0.95]. \tag{5}$$

2.4.2 Firefly Algorithm Setup

In the FA, two parameters are essential for balancing global exploration and local exploitation: the randomization factor αFA and the base attractivenessβ_0. Each one contributes differently to the movement of fireflies, and together they define the algorithm ability to search efficiently across complex, multimodal landscapes. We decided to optimize those two parameters:

Randomization factor:

$$\alpha FA(t) = FIS_{FA}(t_{norm}, D_{FA}, I), \tag{6}$$

With the constraint:

$$\alpha FA(t) \in [0.01, 0.35], \tag{7}$$

Table 3. Performance of DAFACS Baseline on the ten benchmark functions

Functions	Avg	Std
F1	3.29×10^{4}	9.96×10^{3}
F2	6.80×10^{7}	1.41×10^{8}
F3	6.00×10^{1}	4.88×10^{0}
F4	5.73×10^{1}	1.47×10^{2}
F5	3.38×10^{1}	1.74×10^{0}
F6	1.64×10^{-1}	1.61×10^{-2}
F7	2.47×10^{1}	9.98×10^{-1}
F8	-6.57×10^{3}	1.53×10^{2}
F9	7.83×10^{0}	1.81×10^{-1}
F10	6.24×10^{-3}	6.30×10^{-4}

Table 4. Performance of the DAFACS hybrid when Type-1 fuzzy logic is used to adapt the main control parameters

Functions	Avg	Std
F1	1.46×10^{5}	5.47×10^{5}
F2	1.30×10^{1}	8.52×10^{0}
F3	1.94×10^{1}	2.97×10^{0}
F4	4.93×10^{1}	2.26×10^{2}
F5	3.37×10^{1}	2.23×10^{0}
F6	1.60×10^{-1}	1.39×10^{-2}
F7	1.98×10^{1}	3.38×10^{0}
F8	-8.69×10^{3}	3.06×10^{2}
F9	1.68×10^{0}	3.11×10^{-1}
F10	6.11×10^{-3}	7.51×10^{-4}

Base Attractiveness

$$\beta_0(t) = FIS_{FA}(t_{norm}, D_{FA}, I), \tag{8}$$

Restricted to:

$$\beta_0 \in [0.60, 1.60]. \tag{9}$$

2.4.3 Cuckoo Search Algorithm Setup

In the Cuckoo Search (CS) algorithm, two parameters play critical roles in balancing global exploration and local exploitation: the Lévy step size αCS and the abandonment probabilityP_a.

These parameters determine how new solutions are generated and how aggressively the population evolves, directly influencing convergence speed and the ability to escape local optima:

Lévy step size:

$$\alpha CS(t) = FIS_{CS}(t_{norm}, D_{CS}, I) \tag{10}$$

With:

$$\alpha CS \in [0.001, 0.08] \tag{11}$$

Abandonment probability:

$$P_a(t) = FIS_{CS}((t_{norm}, D_{CS}, I) \tag{12}$$

Bounded by:

$$P_a \in [0.10, 0.35] \tag{13}$$

3 Result Analysis

In this section, we report the empirical evaluation of the multi-swarm hybrid composed of DA, FA, and CS. The results are structured to highlight how information is exchanged among the constituent swarms, how the main control parameters are configured, and how the hybrid scheme enhances solution quality relative to the individual algorithms.

Table 3 reports the performance of the proposed hybrid algorithm on the ten benchmark functions F1-F10. For each function, the table shows the average best fitness value (AVG) and the corresponding standard deviation (STD) over repeated independent runs. Overall, the hybrid approach is able to obtain competitive objective values on all problems, while maintaining a reasonably low variability in most cases.

For F1 and F2 the mean fitness values are

3.29×10^{4} and 6.80×10^{7}, respectively, with standard deviations of 9.96×10^{3} and 1.41×10^{8}.

These results indicate that the hybrid optimizer can approach good-quality solutions on these relatively difficult landscapes, although the relatively large STD values suggest that the convergence behavior is more sensitive to the initial population on these functions. In contrast, for

ISSN 2007-9737

F3 the algorithm obtains a mean value of 6.00×10^{1} with a much smaller STD of 4.88×10^{0}, reflecting a more stable convergence pattern.

The performance on F4 - F7 shows that the hybrid method can handle functions with different scales and characteristics. For F5 and F7, the mean values (3.38×10^{1} and 2.47×10^{1}) combined with small standard deviations (1.74×10^{0} and 9.98×10^{-1}) indicate that the algorithm consistently converges to similar high-quality regions of the search space. Although F4 presents a higher dispersion (STD = 1.47×10^{2}), the hybrid strategy is still able to keep the average solution around 5.73×10^{1}, showing that the cooperation among the three metaheuristics helps to escape poor local optima even in more irregular landscapes. For F6, the hybrid obtains a very low average fitness of 1.64×10^{-1} with a STD of 1.61×10^{-2}, which evidences precise and robust convergence toward the global basin of attraction. The last three functions, F8 – F10, further highlight the exploitation capability of the proposed approach. On F8, the mean objective value reaches -6.57×10^{3} with a relatively small standard deviation of 1.53×10^{2}, showing that the algorithm can reliably locate deep minima in highly non-linear search spaces. For F9 and F10, the average results (7.83×10^{0} and 6.24×10^{-3}) and very low STD values (1.81×10^{-1} and 6.30×10^{-4}) confirm that the hybrid design is able to finely refine the solutions once a promising region has been identified. In summary, the statistics in Table 3 indicate that the cooperative interaction between DA, FA and CS yields a good balance between exploration and exploitation, leading to accurate and relatively stable solutions across all ten benchmark functions.

Table 4 summarizes the performance of the DAFACS hybrid when Type-1 fuzzy logic is used to adapt the main control parameters. As before, the table reports the average best fitness (AVG) and the standard deviation (STD) over multiple independent runs on the benchmark set F1 – F10

For the unimodal and relatively smooth functions, the Type-1 fuzzy DAFACS exhibits accurate and stable behavior. On F5, F6, F9, and F10, the algorithm attains low mean objective values (e.g., 3.37×10^{1} on F5 and 6.11×10^{-3} on F10) with very small standard deviations, indicating consistent convergence across runs. In particular, the tiny dispersion observed on F6, F9, and F10 shows that the fuzzy parameter adaptation is effective at fine-tuning the exploitation phase once the hybrid optimizer has located a promising region.

The behavior on multimodal and more irregular landscapes also highlights relevant strengths. For F3 and F7, the hybrid achieves mean values of 1.94×10^{1} and 1.98×10^{1}, respectively, with moderate variability, suggesting that the interaction between DA, FA, and CS guided by Type-1 fuzzy rules helps maintain a good balance between exploration and exploitation.

On F8, the algorithm consistently reaches deep minima, with an average of -8.69×10^{3} and a relatively small standard deviation of 3.06×10^{2}, which confirms the capability of the fuzzy-driven hybrid to escape shallow local optima and to exploit complex search spaces. Although the dispersion is larger for some difficult functions such as F1 - F2, and F4, the corresponding mean values remain competitive, showing that the fuzzy-controlled DAFACS can still identify high-quality solutions even when the landscape induces a more stochastic convergence pattern. Overall, these results indicate that incorporating Type-1 fuzzy logic into the DAFACS framework enhances the robustness of the search process and improves the reliability of the hybrid optimizer on a broad range of benchmark functions.

Table 5 reports the performance of the DAFACS hybrid when interval Type-2 fuzzy logic is used to drive the parameter adaptation mechanism. As in the previous configurations, the table summarizes the average best fitness value (AVG) and the corresponding standard deviation (STD) over multiple independent runs on the benchmark suite F1 – F10. On several functions, the interval Type-2 fuzzy DAFACS shows strong robustness and accurate convergence. For instance, on F6, F7, and F10 the algorithm attains low mean objective values (1.67×10^{-1}, 2.46×10^{1}, and 6.03×10^{-3}, respectively) together with very small standard deviations. This behavior indicates that the interval Type-2 fuzzy rules are effective at stabilizing the dynamics of the hybrid optimizer and at maintaining consistent performance across different runs, even in the presence of uncertainty in the search process. Similarly, for F3 and F5, the averages remain in the order of 101 with moderate variability, confirming a good balance between

doi: 10.13053/CyS-30-1-6333

Table 5. Performance of the DAFACS hybrid when interval Type-2 fuzzy logic is used to drive the parameter adaptation mechanism

Functions	Avg	Std
F1	1.24×10^{6}	1.08×10^{6}
F2	2.61×10^{1}	9.23×10^{0}
F3	2.63×10^{1}	2.77×10^{0}
F4	6.28×10^{1}	2.26×10^{2}
F5	3.40×10^{1}	2.36×10^{0}
F6	1.67×10^{-1}	1.17×10^{-2}
F7	2.46×10^{1}	9.87×10^{-1}
F8	-7.05×10^{3}	2.33×10^{2}
F9	3.19×10^{0}	1.56×10^{-1}
F10	6.03×10^{-3}	8.84×10^{-4}

Table 6. Performance of the DAFACS hybrid when generalized Type-2 fuzzy logic is employed for parameter adaptation

Functions	Avg	Std
F1	1.59×10^{-1}	1.32×10^{-2}
F2	1.02×10^{1}	2.89×10^{0}
F3	1.96×10^{1}	2.59×10^{0}
F4	6.94×10^{1}	2.38×10^{2}
F5	3.43×10^{1}	2.53×10^{0}
F6	1.61×10^{-1}	1.63×10^{-2}
F7	1.99×10^{1}	1.83×10^{0}
F8	-8.62×10^{3}	2.70×10^{2}
F9	1.73×10^{0}	2.67×10^{-1}
F10	6.13×10^{-3}	8.97×10^{-4}

exploration and exploitation on moderately complex landscapes.

The performance on the more challenging multimodal functions further highlights the strengths of the proposed approach. On F8, the hybrid reaches a deep minimum with an average fitness of -7.05×10^{3} and a relatively small dispersion, showing that the cooperative interaction of DA, FA, and CS, modulated by interval Type-2 fuzzy sets, is capable of guiding the search toward high-quality regions while avoiding premature convergence. Although functions such as F1, F2, and F4 exhibit larger standard deviations, the corresponding mean values are still competitive, suggesting that the additional degrees of freedom introduced by the Type-2 representation allow the algorithm to adapt to heterogeneous landscapes without sacrificing overall solution quality. Overall, these results indicate that the interval Type-2 fuzzy DAFACS configuration provides a robust and flexible optimization framework, capable of handling uncertainty and variability in the search space while preserving the global search capabilities of the underlying hybrid.

Table 6 presents the performance of the DAFACS hybrid when generalized Type-2 fuzzy logic is employed for parameter adaptation. As before, the table reports the average best fitness (AVG) and the corresponding standard deviation (STD) over multiple independent runs on the benchmark functions F1 – F10.

The generalized Type-2 configuration delivers very strong and highly stable results on several functions. On F1 the method achieves an average value of 1.59×10^{-1} with a very small standard deviation of 1.32×10^{-2}, indicating precise convergence and low sensitivity to the initial population. Similarly, for F6, F7, F9, and F10 the algorithm attains low mean objective values (1.61×10^{-1}, 1.99×10^{1}, 1.73×10^{0}, and 6.13×10^{-3}, respectively) with limited dispersion. This consistent behavior suggests that the higher expressiveness of generalized Type-2 fuzzy sets helps to model uncertainty in the search dynamics and to maintain a stable balance between exploration and exploitation.

For moderately complex landscapes such as F2, F3, and F5, the generalized Type-2 DAFACS also provides competitive averages with controlled variability, which confirms that the fuzzy adaptation mechanism can effectively guide the cooperative interaction among DA, FA, and CS across different problem characteristics. On the more challenging multimodal functions, the hybrid continues to show good global search capabilities. In particular, on F8 the optimizer finds deep minima with an average value of -8.62×10^{3} and a standard deviation of 2.70×10^{2}, demonstrating that the generalized Type-2 representation allows the swarm to exploit

Table 7. Comparison of the four DAFACS variants

Functions	Best AVG	Best config	Worst AVG	Worst config
F1	1.59×10^{-1}	GT2	1.24×10^{6}	IT2
F2	1.02×10^{1}	GT2	6.80×10^{7}	Baseline
F3	1.94×10^{1}	T1	6.00×10^{1}	Baseline
F4	4.93×10^{1}	T1	6.94×10^{1}	GT2
F5	3.37×10^{1}	T1	3.43×10^{1}	GT2
F6	1.60×10^{-1}	T1	1.67×10^{-1}	IT2
F7	1.98×10^{1}	T1	2.47×10^{1}	Baseline
F8	-8.69×10^{3}	T1	-6.57×10^{3}	Baseline
F9	1.68×10^{0}	T1	7.83×10^{0}	Baseline
F10	6.03×10^{-3}	IT2	6.24×10^{-3}	Baseline

Table 8. Comparison of Z Test

Functions	Z (Baseline vs Type-1)	Z Baseline vs Interval T2)	Z(Baseline vs Generalized T2)	Winner
F1	n.s.	-6.12	18.09	Generalized T2
F2	2.64	2.64	2.64	Generalized T2
F3	38.93	32.89	40.05	Type-1
F4	n.s.	n.s.	n.s.	Type-1
F5	n.s.	n.s.	n.s.	Type-1
F6	n.s.	n.s.	n.s.	Type-1
F7	7.62	n.s.	12.61	Type-1
F8	33.94	9.43	36.18	Type-1
F9	93.61	106.36	103.58	Type-1
F10	n.s.	n.s.	n.s.	Interval T2

promising regions while still preserving enough diversity to avoid premature convergence. Overall, these results indicate that the generalized Type-2 fuzzy DAFACS variant is a robust and flexible optimizer, capable of handling heterogeneous and uncertain search spaces while delivering accurate and repeatable solutions across the benchmark set.

A summary in Table 7 shows a comparison of the four DAFACS variants (baseline hybrid without fuzzy logic, Type-1 fuzzy DAFACS, interval Type-2, and generalized Type-2) shows clear advantages of incorporating fuzzy parameter adaptation.

Overall, the Type-1 fuzzy DAFACS achieves the best mean fitness on 7 out of 10 functions (F3,F4,F5,F6,F7,F8,F9F3,F4,F5,F6,F7,F8,F9).For these problems, its average values are consistently lower than those of the baseline and the Type-2 versions. For example, on F7F7 and F9F9 it reduces the mean error from 2.47×1012.47×101 and 7.83×1007.83×100 (baseline) to 1.98×1011.98×101 and 1.68×1001.68×100, respectively. On the multimodal function F8F8, Type-1 obtains the deepest minimum with −8.69×103−8.69×103, outperforming both the baseline and the Type-2 configurations. These results indicate that Type-1 fuzzy rules provide a very effective and robust balance between exploration and exploitation for most of the benchmark set. The generalized Type-2 DAFACS is particularly strong on the more challenging functions F1F1 and F2F2, where it achieves the lowest means of all methods. It reduces the average value on F1F1 down to 1.59×10−11.59×10−1, whereas the baseline and interval Type-2 versions remain in the order of 104104–106106. Similarly, on F2F2 it reaches 1.02×1011.02×101, clearly better than both the baseline (6.80×1076.80×107) and the interval Type-2 variant. This suggests that the extra degrees of freedom of generalized Type-2 sets are especially beneficial for highly sensitive or ill-conditioned landscapes. However, the same configuration yields the worst averages on F4F4 and F5F5, indicating that its added flexibility may sometimes over-adapt on simpler structures.

The interval Type-2 DAFACS does not dominate the majority of benchmarks, but it achieves the best result on F10F10 with a mean of 6.03×10−36.03×10−3, slightly improving over both the baseline and Type-1 configurations. This indicates that interval Type-2 uncertainty modeling can provide fine-grained improvements on high-precision functions with very small objective values.

The baseline DAFACS hybrid (without fuzzy logic) never attains the best mean on any function and appears as the worst performer on several problems (F2, F3, F7, F8, F9, F10). In particular,

ISSN 2007-9737

on F2 and F3 its averages are orders of magnitude worse than those of the fuzzy-enhanced variants. This confirms that fuzzy-based parameter adaptation is not only helpful but necessary to fully exploit the potential of the DAFACS hybrid.

In summary, the comparison shows that: Type-1 fuzzy DAFACS is the most consistently competitive across the benchmark set. Generalized Type-2 fuzzy DAFACS provides large performance gains on the hardest functions (F1,2F1,F2). Interval Type-2 fuzzy DAFACS contributes targeted improvements on specific precision-oriented problems (e.g., F10).

All fuzzy variants clearly outperform the baseline hybrid on most functions, highlighting the effectiveness of fuzzy parameter adaptation in the proposed DAFACS framework.

3.1 Statistical Significance Analysis (Z-test)

Table 8 summarizes the Z-test results comparing the baseline DAFACS hybrid against the three fuzzy variants (Type-1, interval Type-2, and generalized Type-2) on each benchmark function. For every function, the table reports the Z statistic for the pairwise comparison Baseline vs. Fuzzy configuration, but only when the difference is statistically significant at α=0.05(i.e., $|Z|$>1.96); non-significant cases are marked as (“n.s”). Positive Z values indicate that the fuzzy variant outperforms the baseline (lower mean fitness), whereas negative Z values indicate that the baseline performs better.

In the last column, the method achieving the minimum average objective value among all four configurations is reported as the best performer for each function. The results show that the generalized Type-2 DAFACS clearly dominates on F1 and F2, while the Type-1 fuzzy DAFACS is the best performer on most of the remaining functions (F3, F4, F5, F6, F7, F8, F9). The interval Type-2 variant achieves the best result on F10. Moreover, for several functions (e.g., F3, F7, F8, and F9), the absolute Z values are very large, confirming that the improvements of the fuzzy-enhanced DAFACS variants over the baseline are statistically significant and not due to random fluctuations.

4 Conclusions

In this work, we proposed a cooperative hybrid optimizer that couples the Dragonfly Algorithm, the Firefly Algorithm and Cuckoo Search through a shared global best solution and fuzzy-based parameter adaptation. Type-1, Interval Type-2 and General Type-2 fuzzy controllers were used to adjust key parameters of the three metaheuristics during the run, so that exploration and exploitation are not kept fixed but are modified according to the current state of the search. The approach was evaluated on a set of ten standard benchmark functions that include both unimodal and multimodal landscapes.

The experiments show that the baseline hybrid (without fuzzy adaptation) already exhibits competitive behavior compared with the individual algorithms, but the inclusion of fuzzy controllers brings a consistent improvement in most cases. The Type-1 configuration tends to reduce the average error and smooth the convergence curve, while the Interval Type-2 variant offers a more stable performance when the landscape is more irregular. Among all configurations, the General Type-2 controller usually provides the lowest mean errors and the most reliable convergence patterns on the majority of the tested functions. The statistical Z-tests reported in the tables confirm that these differences are statistically significant for several of the benchmark problems, indicating that the improvements are not due to random fluctuations.

From a methodological perspective, the results suggest that both cooperation and uncertainty handling are important for dynamic parameter control. The three metaheuristics contribute different search behaviors: DA introduces structured swarm interactions, FA improves local refinement around promising regions, and CS enhances long-range moves that help escape local minima. The fuzzy layers modulate the relative influence of these behaviors based on simple indicators of search progress, such as iteration stage and recent improvement, leading to a more adaptive balance between exploration and exploitation over time.

This study has some natural limitations. The evaluation was restricted to a limited set of continuous benchmark functions and to a fixed experimental setting; other families of problems, such as constrained, noisy or dynamic optimization tasks, were not considered here. Future work will therefore focus on extending the fuzzy-enhanced hybrid to high-dimensional scenarios, constrained engineering design problems and application domains such as medical decision support or classification tasks. It would also be interesting to study more complex General Type-2 designs and more efficient implementations, for example by using surrogate models or GPU-based computation, in order to reduce the computational cost and make the approach more attractive for large-scale real-world systems.

References

1. **Yang, X.-S. (2021).** Nature-inspired optimization algorithms. Elsevier, 1–310. DOI: 10.1016/C2019-0-00859-0
2. **Talbi, E.-G. (2022).** Metaheuristics: From design to implementation. Wiley, 1–450. DOI: 10.1002/9780470496916
3. **Mirjalili, S. (2016).** Dragonfly algorithm: A new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems. Neural Computing and Applications, 27(4), 1053–1073. DOI: 10.1007/s00521-015-1920-1
4. **Yang, X.-S. (2010).** Firefly algorithm, stochastic test functions and design optimization. International Journal of Bio-Inspired Computation, 2(2), 78–84. DOI: 10.1504/IJBIC.2010.032124
5. **Yang, X.-S., Deb, S. (2010).** Cuckoo search via Lévy flights. IEEE World Congress on Nature & Biologically Inspired Computing, 210–214. DOI: 10.1109/NABIC.2009.5393690
6. **Lee, J., Yoon, Y., Kim, J., Kim, Y.-H. (2024).** Metaheuristic-Based Feature Selection Methods for Diagnosing Sarcopenia with Machine Learning Algorithms. Biomimetics, 9(3), 179.DOI: 10.3390/biomimetics9030179
7. **Xu, M., Cao, L., Lu, D., Hu, Z., Yue, Y. (2023).** Application of Swarm Intelligence Optimization Algorithms in Image Processing: A Comprehensive Review of Analysis, Synthesis, and Optimization. Biomimetics, 8(2), 235. DOI: 10.3390/biomimetics8020235
8. **Castillo, O., Melin, P., Valdez, F., González, C., García, M., Mancilla, A., Cortes-Antonio, P., Soria, J. (2025).** A Review on the Role of Fuzzy Logic in Hybrid Intelligent Systems. Computación y Sistemas, 29(3), 1723–1740.DOI: 10.13053/CyS-29-3-5897
9. **Wolpert, D.H., Macready, W.G. (1997).** No free lunch theorems for optimization. IEEE Transactions on Evolutionary Computation, 1(1), 67–82. DOI: 10.1109/4235.585893
10. **Mendoza, O., Melin, P., Licea, G. (2009).** A hybrid approach for image recognition combining type-2 fuzzy logic, modular neural networks and the Sugeno integral. Information Sciences, 179(13), 2078–2101.DOI: 10.1016/j.ins.2008.11.018
11. **Houssein, E. H., Saber, E., Wazery, Y. M., Ali, A. A. (2022).** Swarm Intelligence Algorithms-Based Machine Learning Framework for Medical Diagnosis: A Comprehensive Review. **E. H. Houssein, M. A. Elaziz, D. Oliva, L. Abualigah,** editors, Integrating Meta-Heuristics and Machine Learning for Real-World Optimization Problems (Studies in Computational Intelligence, vol. 1038, pp. 85–106). Springer. DOI: 10.1007/978-3-030-99079-4_4
12. **Boussaïd, I., Lepagnot, J., Siarry, P. (2013).** A survey on optimization metaheuristics. Information Sciences, 237, 82–117. DOI: 10.1016/j.ins.2013.02.041
13. **Kao, Y.-T., Zahara, E. (2008).** A hybrid genetic algorithm and particle swarm optimization for multimodal functions. Applied Soft Computing, 8(2), 849–857.
14. **Hassan, R., Cohanim, B., de Weck, O., Venter, G. (2005).** A comparison of particle swarm optimization and the genetic algorithm. 46th AIAA/ASME/ASCE/AHS/ASC Structures Conference, 1–13. DOI: 10.2514/6.2005-1897
15. **Li, X., Yin, M. (2015).** Modified cuckoo search algorithm with self adaptive parameter method. Information Sciences, 298, 80–97. DOI: 10.1016/j.ins.2014.11.042
16. **Gandomi, A. H., Yang, X.-S., Alavi, A. H. (2011).** Mixed variable structural optimization

using Firefly Algorithm. Computers & Structures, 89(23–24), 2325–2336. DOI: 10.1016/j.compstruc.2011.08.002

17. **Hussain, K., Salleh, M. N. M., Cheng, S., Shi, Y. (2019).** Metaheuristic research: A comprehensive survey. Artificial Intelligence Review, 52, 2191–2233. DOI: 10.1007/s10462-017-9605-z
18. **Skanderová, L. (2022).** Self-organizing migrating algorithm: review, improvements and comparison. Artificial Intelligence Review, 56(1), 101–172. DOI: 10.1007/s10462-022-10167-8
19. **Reyes-Sierra, M., Coello Coello, C.A. (2006).** Multi-objective particle swarm optimizers: A survey of the state-of-the-art. International Journal of Computational Intelligence Research, 2(3), 287–308. DOI: 10.5019/j.ijcir.2006.68
20. **Castillo, O., Amador-Angulo, L. (2018).** A generalized type-2 fuzzy logic approach for dynamic parameter adaptation in bee colony optimization applied to fuzzy controller design. Information Sciences, 460–461, 476–496. DOI: 10.1016/j.ins.2017.10.032
21. **Miramontes, I., Melin, P. (2022).** Interval Type-2 Fuzzy Approach for Dynamic Parameter Adaptation in the Bird Swarm Algorithm for the Optimization of Fuzzy Medical Classifier. Axioms, 11(9), 485. DOI: 10.3390/axioms11090485
22. **Juang, Y.-T., Tung, S.-L., Chiu, H.-C. (2011).** Adaptive fuzzy particle swarm optimization for global optimization of multimodal functions. Information Sciences, 181(20), 4539–4549. DOI: 10.1016/j.ins.2010.11.025
23. **Olivas, F., Valdez, F., Castillo, O., Melin, P. (2016).** Dynamic parameter adaptation in particle swarm optimization using interval type-2 fuzzy logic. Soft Computing, 20(3), 1057–1070. DOI: 10.1007/s00500-014-1567-3
24. **Peraza, C., Valdez, F., Castro, J.R., Castillo, O. (2018).** Fuzzy Dynamic Parameter Adaptation in the Harmony Search Algorithm for the Optimization of the Ball and Beam Controller. Advances in Operations Research, 2018, Article ID 3092872, 1–16. DOI: 10.1155/2018/3092872
25. **Segura, C., Coello Coello, C.A., Segredo, E., Hernández-Aguirre, A. (2016).** A novel diversity-based replacement strategy for evolutionary algorithms. IEEE Transactions on Cybernetics, 46(12), 3233–3246. DOI: 10.1109/TCYB.2015.2501726
26. **Tanabe, R., Fukunaga, A. (2013).** Success-history based parameter adaptation for differential evolution. 2013 IEEE Congress on Evolutionary Computation, 71–78. DOI: 10.1109/CEC.2013.6557555
27. **Wagner, C., Hagras, H. (2010).** Toward General Type-2 Fuzzy Logic Systems Based on zSlices. IEEE Transactions on Fuzzy Systems, 18(4), 637–660.DOI: 10.1109/TFUZZ.2010.2045386
28. **Amador-Angulo, L., Castillo, O., Melin, P., Geem, Z. W. (2024).** Generalized Type-2 Fuzzy Approach for Parameter Adaptation in the Whale Optimization Algorithm. Mathematics, 12(24), 4031.DOI: 10.3390/math12244031
29. **Valdez, F. (2020).** A review of optimization swarm intelligence-inspired algorithms with type-2 fuzzy logic parameter adaptation. Soft Computing, 24(1), 215–226. DOI: 10.1007/s00500-019-04290-y
30. **Wang, H., Jin, Y., Yao, X. (2017).** Diversity assessment in many-objective optimization. IEEE Transactions on Cybernetics, 47(6), 1510–1522. DOI: 10.1109/TCYB. 2016.2550502
31. **Valdez, F., Castillo, O., Melin, P. (2025).** A Review on Enhanced Evolutionary Algorithms Based on Extensions of Fuzzy Logic Systems. Computación y Sistemas, 29(2), 587–597. DOI: 10.13053/CyS-29-2-5661

Article received on 31/10/2025; accepted on 15/12/2025.
**Corresponding author is Fevrier Valdez.*